Experimental Validation of an End-to-End QKD Encryption Service in MPLS environments

A. Aguado[1], J. Martinez-Mateo[1], V. Lopez[2], D. Lopez[2], M.Peev[3], V. Martin[1]
[1]Center for Computational Simulation - Universidad Politécnica de Madrid. Campus de Montegancedo. Boadilla del Monte, 28660 Madrid. Spain
[2]Telefónica Investigación y Desarrollo, Ronda de la Comunicación s/n 28050 Madrid. Spain
[3]Huawei Technologies Duesseldorf GmbH, Riesstrasse 25, 80992 Munchen. Germany
(a.aguadom@fi.upm.es, vicente@fi.upm.es)

ABSTRACT.— Current network architectures are rapidly evolving towards more dynamic solutions, due to the ever-growing demand of resources from highly-heterogeneous new services. This forces network management systems and protocols to quickly adapt to support new capabilities and security is one of the main concerns. In this work, we define and demonstrate extensions for multi-protocol label switching (MPLS) networks using quantum key distribution (QKD) keys to secure end-to-end (E2E) network services. The proposed solution allows to synchronize key IDs between remote endpoints, as well as to transmit other parameters required for the encryption. Results show how these new services could be integrated in existing operators control plane architectures.

Statement of the problem.— NETWORK services are continuously evolving, moving from a traditional approach, based on proprietary devices performing a fixed function, to more flexible and open solutions, implemented in general purpose hardware and using open standards. SDN allows to decouple the control plane (the management of the network itself) from the data plane (the actual transmission of the data). These networks are more flexible and configurable but their security risks are correspondingly larger and, therefore, security in network infrastructures must be enhanced. QKD can be seen as a new opportunity for operators and infrastructure providers as it can enable the provision of new, high security encryption in end-to-end (E2E) services. This work describes the proposed solution for integrating QKD into network services via MPLS control plane (as its generalized version, GMPLS).

Node architecture and workflow.— In order to enhance current network services providing such new capabilities, we must first define the dependencies in terms of node architecture and control plane requirements. This type of nodes (Fig. 1 left) must:

- Generate keys synchronized with remote peers to perform symmetric encryption (in our case, via QKD).
- Encrypt the outcoming traffic utilizing those keys.
- Be able of performing switching or routing (optional).
- Communicate (northbound intarface) with a network controller.

Similarly, the MPLS agent must implement certain extensions to synchronize with the control plane (workflow and protocol extensions). Our proposed workflow, including those operations and the main messages you be exchanged in a MPLS-enabled network, is shown in Fig. 1 (right). Each step is as indicated below:

- Initially, a QE node should expose its capabilities to the centralized controller. In our case, it will consist on a path computation element -PCE- or more complex architectures (such as the applications-based network operations -ABNO- architecture).
- The request for an E2E QE service could come from the controllers NBI or from the device itself. In our figure, the node 1 sends a path computation request, sending the encryption requirements encapsulated in metrics.
- The path is returned to the source node, which will detect the QE requirements. Upon receipt, this node will extract the key and key ID pair from a QKD system (or key manager).
- The key ID will be forwarded via RSVP path message to the destination node, which will extract the required key from the peer QKD system (or key manager).
- This process will finalise when a RSVP Resv message arrives to the source node, acknowledging the configuration.

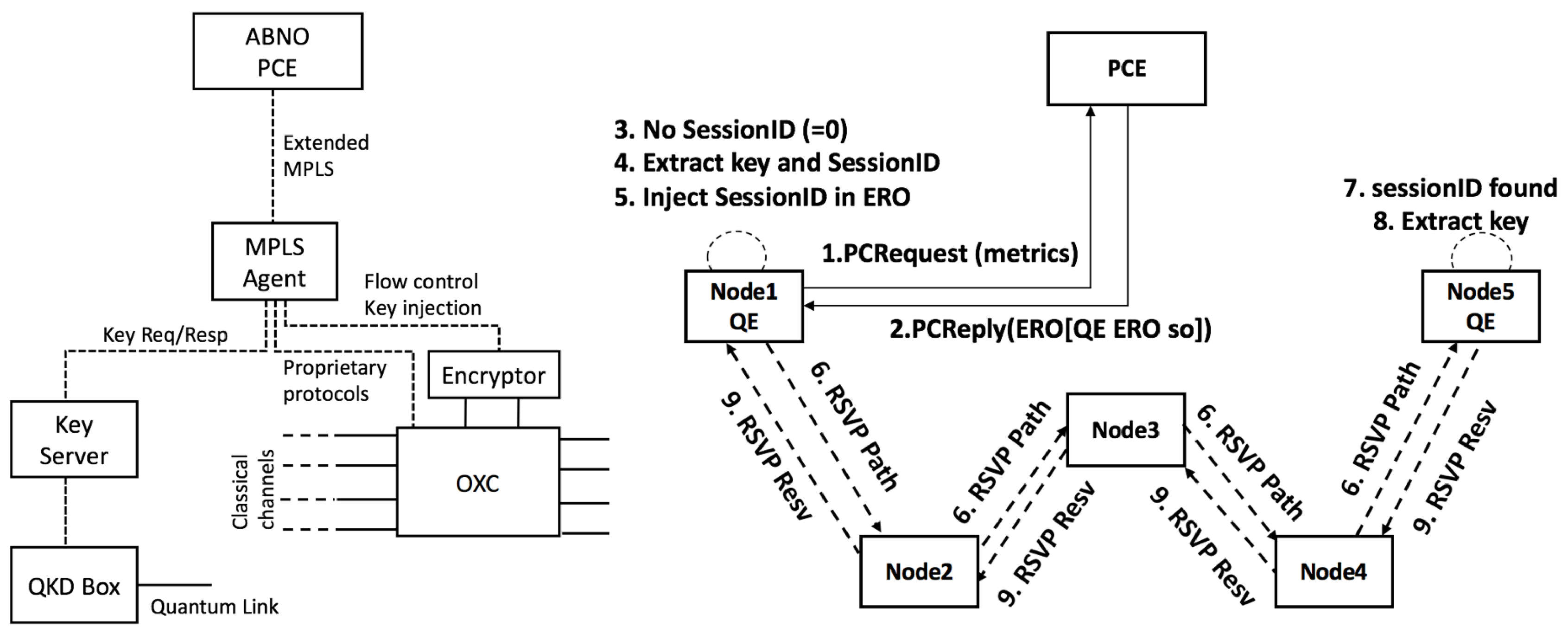

Fig. 1. (Left) Structure of an QE-enabled node, including: an MPLS agent orchestrating the multiple entities and comunicating with the PCE, the QKD domain including the QKD box and the key server, the OXC switch and an encryptor. (Right) MPLS workflow for enabling the QE end-to-end service.

Experimental Results.— The full set of the messages transmitted across the GMPLS control plane network is shown in Fig. 4. The first message is an OSPF update from the fifth node (others are omitted), which contains the router information opaque LSA, with the traffic engineering capable and que QE capable bits set to 1 within the informational capabilities TLV. The second and third messages are the PCRequest and PCReply messages, including the new metrics and the new QE ERO SO (Fig. 3). The remaining messages are the RSVP path and resv messages, used to forward the configuration from the ingress (source) to the engress (destination) nodes of the path.

Finally, Fig. 5 shows how the QE ERO SO is modified by the source node, by including the valid key ID to be used by the encryption path. The interface used to retrived the keys from the QKD systems has been designed using the API specification described in (8). Upon receipt, the destination node gets the valid key ID to retrieve a key from the corresponding QKD system for the encrypted channel.

22.910357	10.1.1.5	224.0.0.5	OSPF	LS Update
30.691575	10.1.1.1	10.1.1.200	PCEP	Path Computation Req
30.733564	10.1.1.200	10.1.1.1	PCEP	Path Computation Rep
30.799866	10.1.1.1	10.1.1.2	RSVP	PATH Message. SESSIO
30.852555	10.1.1.2	10.1.1.3	RSVP	PATH Message. SESSIO
30.892129	10.1.1.3	10.1.1.4	RSVP	PATH Message. SESSIO
30.941130	10.1.1.4	10.1.1.5	RSVP	PATH Message. SESSIO
30.964791	10.1.1.5	10.1.1.4	RSVP	RESV Message. SESSIO
30.974283	10.1.1.4	10.1.1.3	RSVP	RESV Message. SESSIO
30.988739	10.1.1.3	10.1.1.2	RSVP	RESV Message. SESSIO
31.017170	10.1.1.2	10.1.1.1	RSVP	RESV Message. SESSIO

Fig. 2. Set of MPLS messages transmitted for setting up the QE end-to-end service.

Protocol requirements.— Handling the setting up of this new type of services requires defining and implementing extensions to perform three types of operations: features dissemination, configuration and signaling. While the first one is used by the control plane to identify the QE-capable nodes, the others are used to configure the service itself. The proposed extension to disseminate QE capabilities is based on the RFC7770, which defines OSPF extensions for optional router capabilities. This extension, implemented for OSPFv2, uses the router information (RI) opaque link state advertisement (LSA) within an OSPF update message. A single bit is exposed within an informational capabilities TLV, to let the PCE know that it can provide such type of services.

For configuration and signaling, PCEP and RSVP are the more suitable candidates. When a PCReply or a PCInitiate message arrives to a PCC of a network node, this node is in charge to start the signaling process by extracting the ERO from the PCEP message and transmitting it via an RSVP Path message down to the destination node. In our case, it is mandatory to synchronize the QKD-generated key IDs in both sides. While doing this process through a non-standard channel could be done, RSVP is the best candidate to automate the key synchronization process using a standard protocol. It is capable of forwarding the encryption requirements across the path and return a confirmation (Resv message) if the resources (keys) have been reserved, while the signaling process traverses the network. To encapsulate the key ID together with other important information we have created an explicit route object -ERO- subobject -SO-. This structure contains encryption information, such as key lenght, encryption algorithm, key refresh values (if necessary) and layer of encryption. This structure is transmitted and modified on-the-fly between PCE and the network devices to synchronized both ends of the encrypted path.

Before Node 1 (PCRequest) QE ERO Subobject

```
0120  20 00 67 4a 00 00 00 00  00 00 00 00 00 00 00 00
0130  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0140  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0160  00 00 00 00 00 20 02 fc  03 e8 00 0a 05 30 00 10
```

After Node 1 (RSVP Path)

```
00f0  00 00 01 08 0a 01 01 05  20 00 67 4a 4a 0e 75 e8
0100  03 d7 f6 9e 9a 29 a1 0d  1c 7b 31 10 ac c3 95 98
0110  b4 78 9f 4f 0d 0e c1 40  fb ca 46 1d 6c a5 d2 a8
0120  a8 cc f0 d4 95 71 76 7d  31 b6 e0 69 4e a0 10 a0
0130  95 89 98 eb df 7d 35 85  e3 e6 05 2f 00 20 02 fc
0140  ff e8 00 0a 00 08 13 01  00 00 00 01 00 0c 0b 07
```

Fig. 3. QE ERO subobject modification, including the valid QKD key ID before and after traversing the source node.

CONCLUSIONS. The combination of novel network paradigms and QKD tech- nologies is just starting. Network infrastructures must quickly adapt to enhance security for their underlying services. In this work we describe and implement extensions to integrate QKD resources into E2E services, all automated across the MPLS control plane. These protocol extensions will be an enabler for operators to offer and capitalize new encrypted network services powered by QKD technologies. QKD as we know it today is just the starting point, but novel models and techniques, such SDN, will allow for the evolution and smooth adaptation these new capabilities and devices.

ACKNOWLEDGMENTS.— This work has been partially supported by the project CVQuCo, TEC2015-70406-R, funded by the Spanish Ministry of Economy and Competitiveness and QUITEMAD+, S2013-IC2801, funded by Comunidad Autónoma de Madrid.